Integer Programming Models for the Geodesic Classification Problem on Graphs

Paulo H. M. Araújo
Federal University of Ceará
Quixadá, Brazil
phmacedoaraujo@ufc.br

Manoel Campêlo
Federal University of Ceará
Fortaleza, Brazil

Ricardo C. Corrêa
Federal Rural University of Rio de Janeiro
Rio de Janeiro, Brazil

Martine Labbé
Free University of Brussels
Brussels, Belgium

ABSTRACT

We study a discrete version of the classical classification problem, to be called geodesic classification problem. It is defined on a graph, where some vertices are initially assigned a class, and the remaining ones must be classified. This vertex partition is grounded on the concept of geodesic convexity on graphs, as a replacement for the Euclidean convexity in the multidimensional space. We propose two integer programming models along with branch-and-cut algorithms to solve them. We also carry out a polyhedral study and run computational experiments to evaluate the proposed approaches.

KEYWORDS

classification, geodesic convexity, polyhedral combinatorics

1 INTRODUCTION

Supervised learning stands for an automatic prediction tool widely used in many situations in nowadays information society. In general terms, it denotes a collection of methods that act on partial information in order to infer the structure of the entire universe with respect to a specific target property. Most frequently, the partial information provided consists of a set of samples whose target property assignments are known, as well as some relationship between these samples. In this context, the automatic prediction is performed through the following two-phase procedure: in the initial phase, or *training phase*, a given sample set is analyzed. Each sample consists of an array of encoded attributes that characterize an object of a certain type together with a label that associates a class to the corresponding object. Most commonly, the target property is the partition of the universe of possible objects in two classes. A tacit assumption made at this phase is that there is an underlying pattern associated with the samples of each class that sets them apart from the samples of the other classes. Thus, the purpose of the training phase is to determine a mapping from all possible objects of the considered type into the set of possible classes as an extension of an underlying pattern of the samples. Then, in the second phase, the mapping determined in the training phase is used to respond to queries about the class of objects that do not belong to the sample set.

An optimization problem is usually associated with the training phase. Referred to as *classification problem*, it consists in grouping similar samples to get clusters as internally homogeneous as possible. A wide range of solution methods is available, each depending on the coding of the samples and the criterion adopted to express homogeneity. A prevalent approach, which we call *Euclidean classification*, is to encode the samples as vectors of numerical features in a multidimensional Euclidean space and to assume that the class patterns can be appropriately characterized by convex sets. More precisely, consider that the samples are colored points in $\mathbb{R}^d$, for some $d \geq 1$, the color representing the class of the sample. The goal is to assign a class (color) to every point in $\mathbb{R}^d$ based on the classification of the samples so that the convex hulls of the colors do not intersect (possibly disregarding some samples as discussed below). In this vein, continuous optimization methods, including linear and quadratic programming, have been developed in the last 40 years. See e.g. [9, 16]. More recently, integer linear programming tools started to be used in conjunction with continuous methods, as we can see in [5, 19].

In [1], a new variant of the classification problem was defined. For this purpose, a correspondence between the convexity concepts in discrete and continuous mathematics can be established if we consider the vertex set of a connected graph and the distance between vertices as metric space. Thus, the *geodesic classification* problem is stated in terms of notions of convexity in graphs and assumes the following hypotheses: (i) the universe of objects is a discrete set (which are not necessarily numerical) represented by a similarity graph $G = (V, E)$, connected, where V is the set of all objects, and E gives the pairs of similar objects; (ii) there exists an underlying classes pattern that can be expressed, or at least approximated, by the notion of *geodesic convexity* in graphs [17]; (iii) the sample set may contain an arbitrary number of misclassified objects, called *outliers*, which result from possible sampling errors or due to inherent characteristics of the phenomenon being modeled.

From the mathematical point of view, an outlier is a classified object that leads the underlying pattern of the samples in its class to deviate from the convexity definition. The possible occurrence of outliers poses an additional challenge to any method used to solve the classification problem since they have to be detected and disregarded so that an accurate solution may be found.

The goal is to split the vertex set into classes, based on the classification of the samples and the structure of the similarity graph, in such a way that an error measure in a metric space is minimized [2]. In this paper, we consider the existence of only two classes in the graph and the number of disregarded outliers as the error measure. The classification of the vertices follows a specific notion of *linear separability* with respect to geodesic convexity.

10.48786/alioeuro.2022.15

From the practical point of view, this problem allows encoding object similarities through some reflexive binary relation. This fact benefits many practical applications in big data, specially in two situations that can arise even when objects are modeled as points in an Euclidean space. The first situation occurs when similarities are expressed in terms of symmetric and non-transitive binary relations. Such a relation define an unweighted similarity graph G. A standard example is to consider as similar any two points that are close to each other in an Euclidean space. The second situation, which arises very often when handling multiple models of text corpora, are constituted by symmetric and transitive relations, thus leading G to be a complete graph (recall that G is assumed to be connected). The particularity of this case is that G is edge-weighted, the weight of an edge standing for a degree of similarity between objects. Cosine similarity in text analysis is an example (see [10] for a general tool based on two distinct topic modeling methods). The theoretical results discussed in this paper assume the first type of similarity relation, but they can be extended directly to the second type if the edge weights are considered in the definition of path length and, consequently, of geodesic convexity. Due to its characteristics, applications of the geodesic classification problem are easily found in the fields of data mining and classical statistics. Text and sentiment analyses, community detection in complex networks (such as social networks and networks of citations of scientific articles), historic files similarity prediction, content recommendation in video streaming services, and spam filtering for e-mails constitute examples thereof [11].

Geodesic and Euclidean classification are distinct problems in the following sense. Consider a set V of points in a multidimensional Euclidean space, and the corresponding similarity graph $G = (V, E)$ such that E connects points whose Euclidean distance is smaller than a given threshold. Assuming, as mentioned above, that the samples form a proper subset of V, the Euclidean classification problem consists in partitioning the Euclidean space into two convex subspaces based on the classification of the samples. Although based on the same samples, the geodesic classification problem aims to split set V only. The possible patterns considered in each problem are distinct. The main reason is that the universe in the former case is composed by selected points of the Euclidean space only. Actually, a solution for the geodesic problem is neither a covering nor a partitioning of G in convex sets in the sense studied in [3] and [8]. Instead, this problem can be seen as the combination of a graph convexity problem and the well-known set covering problem [12], as shown by the mathematical model proposed in Section 2.1.

We have introduced the 2-class geodesic classification problem (2-GC) in the conference paper [1], where preliminary results were presented. In this work, we present two new integer formulations for 2-GC along with a branch-and-cut algorithm for each one. Besides, we run several computational experiments with random and realistic instances to evaluate the geodesic convexity approach. The first formulation has a linear number of variables but an exponential number of constraints, whereas the second one is an extended formulation with more variables but a polynomial number of constraints. An interesting feature of the first model is that it expresses the 2-GC problem as a set covering problem. Thus, we can take advantage of well-known results from the literature.

We also study the polytopes associated with each formulation and show some facet-defining inequalities. Most of the derived facet-defining inequalities can be seen as counterparts of those presented by [6] for a polyhedron that models the Euclidean version of the problem. On the other hand, some of them originates from specific properties of the geodesic case. Despite the fact that the geodesic and Euclidean classification problems are distinct, the study of the combinatorial structure of the former may be useful to design solution methods for the latter.

2 GEODESIC CLASSIFICATION PROBLEM

Let V be a finite set of objects that are related through a binary relation described by the connected undirected graph $G = (V, E)$, $|V| = n$. Two distinct objects are *similar* if the corresponding vertices are adjacent in G (analogously, two vertices $v, w \in V, v \neq w$, are called *similar* if $vw \in E$). We also refer to G as *classification graph*, and we adopt the standard terminology used in Bondy and Murty's book [7]. In particular, a *path between* $v, w \in V$ is the sequence $\langle v \rangle$, if $v = w$, or a sequence of distinct vertices $P = \langle v = v_1, v_2, \ldots, v_\ell = w \rangle$ such that $v_i v_{i+1} \in E$ for $i = 1, \ldots, \ell-1$. The *length* of a path is given by its number of edges, *i.e.* the length of $\langle v \rangle$ is zero and that of P is $\ell - 1$.

A *geodesic* between two vertices h and j in G is a shortest path between h and j in the graph and its length is denoted by $\delta(h, j)$. The closed interval $D[h, j]$ is the set of all vertices lying on a geodesic between h and j. Given $S \subseteq V$, $D[S] = \bigcup_{u,v \in S} D[u, v]$. In this case, if $D[S] = S$, then S is a *convex set*. The *convex hull* of S, denoted by $H[S]$, is the smallest convex set containing S. We also denote $D_{hj} = D[h, j] \setminus \{h, j\}$.

For the definition of the 2-class classification problem considered in this work, we are given two nonempty subsets $V_B, V_R \subseteq V$, $V_B \cap V_R = \emptyset$, so that $V_{BR} = V_B \cup V_R$ represents the sample set. The sets V_B and V_R define the *blue class* and *red class* vertices, respectively. The remaining vertices $V_N = V \backslash V_{BR}$ are called *unclassified vertices.*

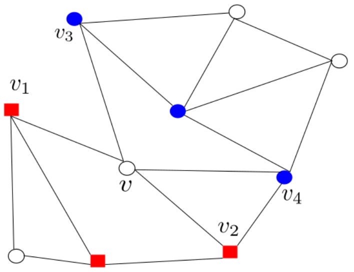


Figure 1: A linearly inseparable classification graph with blue (as solid circles) and red classes (as solid squares) vertices.

In analogy to the Euclidean version of the classification problem, we introduce the following definition of linear separability considering the geodesic convexity.

Definition 2.1. A triple $(A_B \subseteq V_B, A_R \subseteq V_R, A_N \subseteq V_N)$ is *linearly separable (with respect to G)* if

(C1) $H[A_B] \cap A_R = \emptyset$,
(C2) $H[A_R] \cap A_B = \emptyset$, and
(C3) $H[A_B] \cap H[A_R] \cap A_N = \emptyset$

and *linearly inseparable* otherwise (Figure. 1 shows an example of linearly inseparability since vertex $v \in H[V_B] \cap H[V_B] \cap V_N$. If v_1, v_2, v_3 or v_4 is set as an outlier, then it becomes linearly separable).

Subsets A_B and A_R are called a *basis* of the blue and red classes, respectively, whereas $V_{BR} \setminus (A_B \cup A_R)$ are the outliers (in these bases). Each basis consists of non-outlier vertices that *span* on the graph through an operator that expresses the pattern of the corresponding class. In this case, the operator is given by the convex hull. It is worth remarking that considering a vertex $w \in V_{BR}$ as an outlier does not mean removing it from the graph. It only means that w is considered neither red nor blue when calculating the convex hull of the red basis vertices or blue basis vertices. The Geodesic Classification problem for 2 classes becomes:

Problem 1. *2-class Geodesic Classification Problem (2-GC)*:

Given a connected graph $G = (V, E)$, sets of initially classified vertices V_B (blue vertices) and V_R (red vertices), and $V_N = V \setminus (V_{BR})$, find subsets $A_B \subseteq V_B$, $A_R \subseteq V_R$ such that (A_B, A_R, V_N) satisfies (C1), (C2) and (C3), and $|V_{BR}| - |A_B \cup A_R|$ is minimum.

Conditions (C1) and (C2) ensure that if an initially classified vertex i belongs to the convex hull of the non-outlier vertices of its opposite class, then i must be an outlier, i.e., $i \notin A_B \cup A_R$. Since every unclassified vertex needs to be assigned to exactly one class, it must belong to at most one convex hull of non-outliers of the same class. This is guaranteed by Condition (C3). Moreover, we want to find a solution with the minimum number of outliers.

2.1 A Set Covering Formulation for the 2-GC Problem

In this subsection, we formulate the 2-GC problem as a set covering problem of the form

$$\min \left\{ \mathbf{1}^\top \boldsymbol{y} \mid A\boldsymbol{y} \geq \mathbf{1}, \boldsymbol{y} \in \mathbb{B}^n \right\}, \tag{1}$$

where A is a 0-1 matrix and $\mathbf{1}$ is the vector of ones. For a general binary matrix A, (1) is NP-hard [12] and results about valid inequalities and facet-defining properties can be found in [18]. In the particular case of the 2-GC problem, we use a binary variable y_i, for each vertex $i \in V_{BR}$, such that $y_i = 1$ if i is an outlier, and $y_i = 0$ otherwise. Then, using $K(i) \in \{B, R\}$ and $\bar{K}(i) \in \{B, R\} \setminus \{K(i)\}$ to respectively denote the class and the opposite class of vertex $i \in V_{BR}$, matrix A corresponds to the following constraints:

$$\sum_{j \in S \cup \{i\}} y_j \geq 1, \quad i \in V_{BR}, S \subseteq V_{\bar{K}(i)} : i \in H[S], \tag{2}$$

$$\sum_{j \in S \cup T} y_j \geq 1, \quad S \subseteq V_B, T \subseteq V_R : H[S] \cap H[T] \cap V_N \neq \emptyset. \tag{3}$$

This formulation and its corresponding polyhedron will be called ILP1 and P_1 respectively. Proposition 2.2 states that ILP1 is correct.

Proposition 2.2. *Inequalities* (2) *and* (3) *define the feasible solutions of the 2-GC problem.*

It is worth noting that only the constraints of (2) and (3) related to minimal sets S and (S, T), respectively, are necessary to describe P_1. One special case of inequalities (2) that can be separated in polynomial time is obtained when $|S| = 2$ and $i \in H[S]$ is replaced by the stronger condition $i \in D[S]$. We refer to this case as *generalized 3-path inequality* which, in general terms, is written as

$$y_j + y_{j'} + y_i \geq 1, \quad i \in V_{BR}, \{j, j'\} \subseteq V_{\bar{K}(i)} \text{ where } i \in D(j, j'). \tag{4}$$

The following result stems from the fact that each constraint of the integer formulation contains at least two non-null coefficients [4].

Proposition 2.3. *P_1 is full-dimensional.*

Proposition 2.4 states that the bounding inequalities define facets of P_1.

Proposition 2.4. *For every $i \in V_{BR}$, $y_i \geq 0$ and $y_i \leq 1$ are facet-defining for P_1.*

One of the most computationally useful valid inequalities that we found are called *generalized C_4* inequalities and are defined below.

Proposition 2.5 ([1]). *If $v, v' \in V_B$ and $w, w' \in V_R$ are distinct vertices such that $\{v, v'\} \subseteq H[\{w, w'\}]$ and $\{w, w'\} \subseteq H[\{v, v'\}]$, then the following inequalities are facet-defining for P_1:*

$$y_v + y_{v'} + y_w + y_{w'} \geq 2. \tag{5}$$

2.2 A Compact Formulation for the 2-GC Problem

The second integer linear formulation is obtained by including additional variables so as to reduce the number of constraints to a polynomial order. The new binary variables, z, are used to determine if a vertex belongs to the convex hull of the non-outliers of a given class. More precisely, in a 2-GC feasible solution (A_B, A_R), for each $K \in \{B, R\}$ and $i \in V$, we set $z_{K,i} = 1$, if $i \in H[A_K]$, and $z_{K,i} = 0$, otherwise. Thus, the feasible solutions of the new formulation are defined by the binary vectors $\boldsymbol{y} \in \mathbb{B}^n$ and $\boldsymbol{z} \in \mathbb{B}^{2|V|}$ such that

$$y_i \geq z_{\bar{K},i}, \qquad i \in V_K, K \in \{B, R\} \tag{6}$$

$$y_i + z_{K,i} \geq 1, \qquad i \in V_K, K \in \{B, R\} \tag{7}$$

$$z_{B,i} + z_{R,i} \leq 1, \qquad i \in V_N \tag{8}$$

$$z_{K,h} + z_{K,j} - z_{K,i} \leq 1, \quad K \in \{B, R\}, h, i, j \in V : i \in D(h, j) \tag{9}$$

This formulation is denoted ILP2. A big advantage of ILP2 in relation to ILP1 and the formulation in [1] is that we have access to the convex hull of the classes, described by variables z. We could add such variables to the objective function so as to obtain better accuracy results if their corresponding coefficients are set properly.

Next, we show how (2)–(3) and (6)–(9) are related.

Proposition 2.6. *Let $F_1 = \{\boldsymbol{y} \in \mathbb{B}^n \mid$ (2), (3)$\}$ and $F_2 = \{\boldsymbol{y} \in \mathbb{B}^n, \boldsymbol{z} \in \mathbb{B}^{2|V|} \mid$ (6)–(9)$\}$. Then, $F_1 = \operatorname{proj}_{\boldsymbol{y}}(F_2)$.*

We denote by P_2 the polyhedron associated to ILP2. By Proposition 2.6, any valid inequality for P_1 are also valid for P_2. However, the facetness conditions for P_1 are not directly transferred to P_2, even for the bounding constraints. Note that inequalities (5) are also valid for P_2 and remains extremely efficient, but we could only prove its facet-inducing property for P_2 when the C_4 is an induced subgraph.

It is easy to show that P_2 is also full-dimensional. Proposition 2.7 shows the inequalities of ILP2 that define facets.

Proposition 2.7. *The following inequalities are facet-defining for P_2:*

(1) $y_i \leq 1$, *for all* $i \in V_{BR}$;
(2) $z_{K,i} \geq 0$, *for all* $K \in \{B, R\}$ *and* $i \in V_N \cup V_{\bar{K}}$;
(3) $z_{K,i} \leq 1$, *for all* $K \in \{B, R\}$ *and* $i \in V_K$;
(4) $z_{B,i} + z_{R,i} \leq 1$, *for all* $i \in V_N$;

(5) $y_i \geq z_{\bar{K},i}$ *and* $y_i + z_{K,i} \geq 1$*, for all* $i \in V_K$ *and* $K \in \{B, R\}$.

Next, we indicate bounding inequalities that do not define facets of P_2.

Proposition 2.8. *For all* $K \in \{B, R\}$*, the constraints below do not define facets of* P_2*:*

(1) $y_i \geq 0$, $z_{\bar{K},i} \leq 1$*, and* $z_{K,i} \geq 0$*, for all* $i \in V_K$*, and*
(2) $z_{K,i} \leq 1$*, for all* $i \in V_N$.

An *incomplete shortest path* in G is a subsequence $\langle u, \ldots, v\rangle$ of (not necessarily consecutive) vertices of a shortest path between u and v in G. In [14], a generalization of the convexity constraints (9) was presented for the *Path Convex Recoloring* problem. Its application to 2-GC is shown below.

Proposition 2.9. *Let* $\langle u_1, v_1, \ldots, u_t, v_t, u_{t+1}\rangle$ *be an incomplete shortest path in* G *and* $K \in \{B, R\}$*. Then, the* generalized convexity inequality

$$\sum_{\ell=1}^{t+1} z_{K,u_\ell} - \sum_{\ell=1}^{t} z_{K,v_\ell} \leq 1 \tag{10}$$

is valid for P_2.

As a counterpart of the generalized C_4 inequalities (5) given for P_1, we now present valid inequalities for P_2 that also involve variables z for vertices in V_N.

Proposition 2.10. *Let* $S \subseteq V_B \cup V_N$, $|S| = 2$*, and* $T \subseteq V_R \cup V_N$, $|T| = 2$*. If* $S \subseteq H[T]$ *and* $T \subseteq H[S]$*, then the following inequality is valid for* P_2*:*

$$\sum_{i \in S \cap V_N} (1 - z_{B,i}) + \sum_{i \in S \cap V_B} y_i + \sum_{j \in T \cap V_N} (1 - z_{R,j}) + \sum_{i \in T \cap V_R} y_i \geq 2. \tag{11}$$

3 GEODESIC CLASSIFICATION ALGORITHMS

We developed a branch-and-cut algorithm for each formulation. They solve a linear relaxation of the root node, which includes some valid inequalities (cuts) found by separation algorithms, and a lazy constraint approach to find feasible integer solutions. For formulation ILP1, the main steps of our solution method are described in Algorithm 1. Similarly, the main steps of the solution method for formulation ILP2 are described in Algorithm 2.

4 COMPUTATIONAL EXPERIMENTS

In our computational experiments, we aim at analyzing two main aspects: the efficiency of the formulations with/without the derived facet-defining inequalities as well as the accuracy of the provided solution of the classification problem. We used random and realistic instances.

For the analysis of the effects of the valid inequalities used in each algorithm, we tested two other versions of each algorithm, each version obtained by the elimination of Step 3 or Step 4, respectively. The observed results are summarized in Tables 1 and 2. They show the performance of the three tested versions with respect to a standard implementation where both steps 3 and 4 were not applied.

From the results, we could note that the cuts, and specially the inclusion of inequalities (5), were extremely effective, reducing the number of lazy constraints to be added and the running time overall. Also, the lazy constraints approach was very important since it was impracticable to solve the instances without it.

Algorithm 1: ILP1 solving algorithm

```
1 Computation of all $D(h, j)$ sets.
2 Initial cutoff: Since a trivial solution is obtained by taking
    all vertices of a class as outliers, $\min\{|V_B|, |V_R|\}$ is
    provided as a cutoff.
3 Initial model configuration: All generalized $C_4$
    inequalities (5) (with $D(h, j)$ requirement instead of
    $H[\{h, j\}]$) are included in the initial model by
    exhaustive enumeration of all pairs of 2-sized subsets.
    None of the constraints (2)-(3) are used initially.
4 Partial linear relaxation resolution: At the root node of
    the branch-and-cut tree, we solve the linear relaxation of
    the initial model together with the generalized 3-path
    constraints (4) separated as cuts by enumeration.
5 Exact model resolution: Starting from the model obtained
    after Step 4, we add the integrality constraints and solve
    the integer formulation by adding minimal $(S, T)$
    constraints (2)-(3) as lazy constraints.
```

Algorithm 2: ILP2 solving algorithm

```
1 Computation of all $D(h, j)$ sets and inclusion of all
    constraints (6)-(8) in the initial model.
2 Initial cutoff: Since a trivial solution is obtained by taking
    all vertices of a class as outliers, $\min\{|V_B|, |V_R|\}$ is
    provided as a cutoff.
3 Initial model configuration: All generalized $C_4$
    inequalities (5) (with $D(h, j)$ requirement instead of
    $H[\{h, j\}]$) are included in the initial model by
    exhaustive enumeration of all pairs of 2-sized subsets.
    None of the constraints (9) are included initially.
4 Partial linear relaxation resolution: At the root node of
    the branch-and-cut tree, we solve the linear relaxation of
    the initial model together with inequalities (10) and (11)
    (with $D(h, j)$ requirement instead of $H[\{h, j\}]$)
    separated as cuts.
5 Exact model resolution: Starting from the model obtained
    after Step 4, we add the integrality constraints and solve
    the integer formulation by adding (9) as lazy constraints.
```

Algorithm 1	N. of constraints (4)	N. of inequalities (5)	Lazy Const. Reduction	Time Reduction
Step 3 only	0	$2.6\|V\|$	30%	82%
Step 4 only	$14\|V\|$	0	92%	73%
Step 3 and Step 4	$2.9\|V\|$	$2.6\|V\|$	88%	85%

Table 1: Effect of the valid inequalities for ILP1.

Algorithm 2	N. of inequalities (10)	N. of inequalities (5), (11)	Lazy Const. Reduction	Time Reduction
Step 3 only	0	$12\|V\|$	20%	83%
Step 4 only, with (10)	$8\|V\|$	0	85%	11%
Step 3 and Step 4	$5\|V\|$	$17\|V\|$	79%	87%

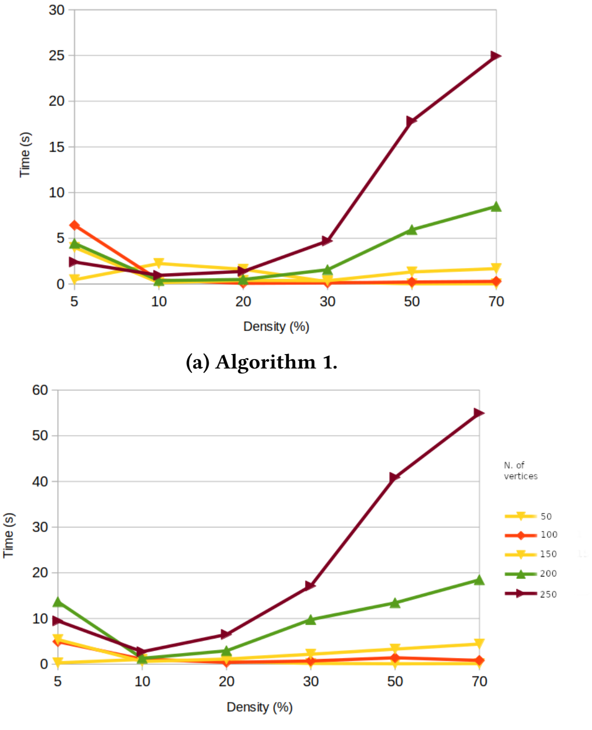
Table 2: Effect of the valid inequalities for ILP2.

4.1 Random Instances

The random instances used in our experiments were categorized by number of vertices $v \in \{50, 100, 150, 200, 250\}$, graph density percentage $d \in \{5, 10, 20, 30, 50, 70\}$ and initially classified vertices percentage $p \in \{60, 80\}$. The initially classified vertices were equally distributed between the blue and the red classes. For each combination v, d and p, we generated 10 random instances, adding up to 600 random instances overall.

For a large part of the random instances, Algorithm 1 had beaten Algorithm 2 in running time. Besides, both algorithms were efficient for small and medium-sized instances, with advantage for Algorithm 1.

Figure 2 – Running time versus density, p = 80.



(a) Algorithm 1.

(b) Algorithm 2.

Figure 2 shows the running times of Algorithm 1 and Algorithm 2 for $p = 80\%$ as a function of the graph density. Overall, the instances with $p = 80\%$ or density between 5% and 20% showed to be the hardest to solve.

4.2 Realistic Instances

To test the developed algorithms for realistic applications, we performed experiments using instances derived from two realistic datasets, namely Parkinson's disease ([15]) and cardiac Single Proton Emission Computed Tomography (SPECT) images ([13]), both available at https://archive.ics.uci.edu/ml/index.php. These datasets are used to generate instances of the Euclidean version of the classification problem, where each point represents the information of a patient to be used to predict new diagnostics.

From each dataset, we derived two groups of 10 instances each for the Euclidean classification problem. An instance in the first (resp. second) group is obtained by randomly choosing 20% (resp. 30%) of the points to form the validation set (points used to check the accuracy of a classification algorithm). Then, each instance is transformed into a classification graph (instance of 2-GC) by using the transformation suggested by [20], where each point becomes a vertex. In particular, the validation points correspond to the initially unclassified vertices. As a way to evaluate the class prediction accuracy of our algorithms, we run the well-known *SVM* and *MLP* Euclidean classification algorithms on the Euclidean instances as well as Algorithm 1, Algorithm 2, and the algorithm of [1] on the corresponding 2-GC instances.

Table 3: Properties of each set of instances.

Instance	n	m	density	diam	minDg	maxDg
parkinsons	195	1097	5	10	1	18
spectf	267	1826	5	8	1	36

Table 4: Algorithm of [1], Algorithm 1, Algorithm 2, *SVM*, and *MLP* comparison for Parkinson's instances with p = 70%.

Instance	$T_{ILPa}(s)$	$T_{ILP2}(s)$	$T_{ILP1}(s)$	$Acu_{GC}(\%)$	$Acu_{SVM}(\%)$	$Acu_{MLP}(\%)$
parkinsons-p70-1	695.23	**23.44**	1152.90	**86.21**	63.79	**86.21**
parkinsons-p70-2	14.72	**1.98**	244.76	68.97	68.97	**74.14**
parkinsons-p70-3	123.60	**18.65**	-	77.59	77.59	**81.03**
parkinsons-p70-4	50.88	**1.78**	53.83	**70.69**	**70.69**	**70.69**
parkinsons-p70-5	31.96	**12.64**	35.28	**74.14**	25.86	72.41
parkinsons-p70-6	324.11	**88.87**	-	**75.86**	**75.86**	72.41
parkinsons-p70-7	34.30	**20.96**	392.59	**79.31**	77.59	77.59
parkinsons-p70-8	83.09	**19.83**	-	**75.86**	24.14	**75.86**
parkinsons-p70-9	373.53	**28.23**	-	82.76	**84.48**	81.03
parkinsons-p70-10	67.56	**17.32**	320.20	75.86	**82.76**	75.86
AVERAGE	179.89	**23.37**	-	**76.72**	65.17	**76.72**

The properties of each set of instances is shown in Table 3. The columns are: number of vertices in the classification graph n; number of edges in the classification graph m; density of the classification graph *dens*; diameter of the classification graph *diam*; minimum degree of the classification graph *minDg*; maximum degree of the classification graph *maxDg*.

Tables 4 and 5 show the results for Parkinson and SPECTF instances with $p = 70\%$ ($p = 80\%$ is omitted). The columns are: running time of the algorithm of [1] $T_{ILPa}(s)$; running time of Algorithm 1 $T_{ILP1}(s)$; running time of Algorithm 2 $T_{ILP2}(s)$; accuracy of the geodesic method $Acu_{GC}(\%)$ (the best among our three methods); accuracy of the *SVM* method $Acu_{SVM}(\%)$; accuracy of the *MLP* method $Acu_{MLP}(\%)$.

Regarding the 20 Parkinson's disease instances, our approach obtained the best accuracy in 10 of them, while 9 and 11 were the scores for *SVM* and *MLP*, respectively. For the 20 SPECT instances, the 2-GC approach presented the best accuracy in 18 instances, while *SVM* and *MLP* did it in 2 and 14 instances, respectively. On average, 2-GC also got the best accuracy, slightly better than the one by *MLP*. Overall, the results show that the accuracy of the 2-GC approach was the best for 28 instances, while *SVM* was the best for only 11 and *MLP* for 25, out of 40 instances.

Comparing the running time of the three algorithms for the geodesic classification in the realistic instances, we note that Algorithm 2 greatly surpasses the other two algorithms for the Parkinson's instances. On the other hand, for the SPECTF instances, the Algorithm of [1] and Algorithm 1, which on average have equivalent results, outperform Algorithm 2.

5 CONCLUDING REMARKS

In this work, we studied and related the polyhedra associated with two new integer formulations that we proposed, giving some important valid inequalities and facet-defining conditions. From the computational experiments, it was clear that the family of facet-defining inequalities called generalized C_4 inequalities was extremely efficient. They are related to a linearly inseparable structure that is not possible to appear in the Euclidean space. The results also show that the proposed solution methods are very promising since the proposed algorithms proved to be

Table 5: Algorithm of [1], Algorithm 1, Algorithm 2, *SVM*, and *MLP* comparison for *SPECT* instances with $p = 70\%$.

Instance	$T_{ILPa}(s)$	$T_{ILP2}(s)$	$T_{ILP1}(s)$	$Acu_{GC}(\%)$	$Acu_{SVM}(\%)$	$Acu_{MLP}(\%)$
spectf-p70-1	**0.17**	2.05	0.25	**83.75**	72.50	**83.75**
spectf-p70-2	0.32	0.79	**0.26**	**78.75**	73.75	**78.75**
spectf-p70-3	**0.15**	1.50	0.16	**80.00**	75.00	**80.00**
spectf-p70-4	0.15	1.65	**0.13**	**76.25**	67.50	75.00
spectf-p70-5	0.18	**0.06**	0.20	**81.25**	41.25	**81.25**
spectf-p70-6	0.08	**0.06**	0.13	**80.00**	62.50	**80.00**
spectf-p70-7	0.20	1.78	**0.19**	**78.75**	51.25	**78.75**
spectf-p70-8	**0.21**	2.60	0.23	**83.75**	60.00	**83.75**
spectf-p70-9	**0.59**	0.75	0.79	**88.75**	87.50	86.25
spectf-p70-10	0.12	**0.06**	0.14	**78.75**	76.25	72.50
AVERAGE	**0.22**	1.13	0.25	**81.00**	66.75	80.00

very efficient in running time and accuracy, even for medium-sized instances. The prediction accuracy of the geodesic approach showed to be stable and as good as such classic linear separation algorithms for the multidimensional space. Therefore, it seems that the analogy performed to transform the Euclidean convexity method into a geodesic convexity method on graphs was successful.

As future works, we intend to carry on a deeper polyhedral study and try new objective functions to improve accuracy.

ACKNOWLEDGMENTS

Partially supported by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001 and the Brazilian agencies CAPES, CNPq (Proc. 305264/2016-8, Proc. 443747/2014-8) and FUNCAP (PNE-0112-00061.01.00/16).

REFERENCES

[1] P. H. M. Araújo, M. Campêlo, R. C. Corrêa, and M. Labbé. 2019. The Geodesic Classification Problem on Graphs. *Electronic Notes in Theoretical Computer Science* 346 (2019), 65 – 76. https://doi.org/10.1016/j.entcs.2019.08.007 The proceedings of LAGOS 2019.

[2] Boris Aronov, Delia Garijo, Yurai Núñez-Rodríguez, David Rappaport, Carlos Seara, and Jorge Urrutia. 2012. Minimizing the error of linear separators on linearly inseparable data. *Discrete Applied Mathematics* 160, 10 (2012), 1441 – 1452. https://doi.org/10.1016/j.dam.2012.03.009

[3] D. Artigas, S. Dantas, M. C. Dourado, and J. L. Szwarcfiter. 2011. Partitioning a graph into convex sets. *Discrete Mathematics* 311, 17 (2011), 1968–1977. https://doi.org/10.1016/j.disc.2011.05.023

[4] E. Balas and S. M. Ng. 1989. On the set covering polytope: I. All the facets with coefficients in {0, 1, 2}. *Mathematical Programming* 43, 1 (01 Jan 1989), 57–69. https://doi.org/10.1007/BF01582278

[5] D. Bertsimas and R. Shioda. 2007. Classification and Regression via Integer Optimization. *Operations Research* 55, 2 (2007), 252–271. https://doi.org/10.1287/opre.1060.0360

[6] M. Blaum, R. C. Corrêa, J. Marenco, I. Koch, and M. Mydlarz. 2019. An integer programming approach for the 2-class single-group classification problem. *Latin & American Algorithms, Graphs and Optimization Symposium* (2019).

[7] J. A. Bondy and U. S. R. Murty. 2008. *Graph Theory.* Springer.

[8] R. Buzatu and S. Cataranciuc. 2015. Convex graph covers. *The Computer Science Journal of Moldova* 23, 3 (2015), 251–269. http://www.math.md/publications/csjm/issues/v23-n3/11974/

[9] C. Cortes and V. Vapnik. 1995. Support-Vector Networks. In *Machine Learning.* 273–297.

[10] PATRICIA J. CROSSNO, ANDREW T. WILSON, TIMOTHY M. SHEAD, IV WARREN L. DAVIS, and DANIEL M. DUNLAVY. 2013. TOPICVIEW: VISUAL ANALYSIS OF TOPIC MODELS AND THEIR IMPACT ON DOCUMENT CLUSTERING. *International Journal on Artificial Intelligence Tools* 22, 05 (2013), 1360008–1 – 1360008–36.

[11] L. Hong and B. D. Davison. 2010. Empirical Study of Topic Modeling in Twitter. In *Proceedings of the First Workshop on Social Media Analytics* (Washington D.C., District of Columbia) *(SOMA '10).* ACM, New York, NY, USA, 80–88. https://doi.org/10.1145/1964858.1964870

[12] R. M. Karp. 1972. *Reducibility among Combinatorial Problems.* Springer US, Boston, MA, 85–103. https://doi.org/10.1007/978-1-4684-2001-2_9

[13] L. A. Kurgan, K. J. Cios, R. Tadeusiewicz, M. Ogiela, and L. S. Goodenday. 2001. Knowledge discovery approach to automated cardiac SPECT diagnosis. *Artificial intelligence in medicine* 23, 2 (Oct 2001), 149–169. https://doi.org/10.1016/S0933-3657(01)00082-3

[14] K.R.P.S. Lima. 2011. *Recoloração Convexa de Caminhos.* Ph. D. Dissertation. PhD thesis, Instituto de Matemática e Estatística da Universidade de São Paulo.

[15] M. A. Little, P. E. McSharry, S. J. Roberts, D. AE Costello, and I. M. Moroz. 2007. Exploiting Nonlinear Recurrence and Fractal Scaling Properties for Voice Disorder Detection. *BioMedical Engineering OnLine* 6, 1 (26 Jun 2007), 23. https://doi.org/10.1186/1475-925X-6-23

[16] P. M. Pardalos and P. Hansen. 2008. *Data mining and mathematical programming.* Vol. 45. American Mathematical Society, Providence, RI.

[17] I. M. Pelayo. 2013. *Geodesic Convexity in Graphs.* Springer-Verlag, New York.

[18] M. Sánchez-García, M. I. Sobrón, and B. Vitoriano. 1998. On the set covering polytope:Facets with coefficients in {0, 1, 2, 3}. *Annals of Operations Research* 81 (01 Jun 1998), 343–356. https://doi.org/10.1023/A:1018969410431

[19] G. Xu and L. G. Papageorgiou. 2009. A Mixed Integer Optimisation Model for Data Classification. *Comput. Ind. Eng.* 56, 4 (May 2009), 1205–1215. https://doi.org/10.1016/j.cie.2008.07.012

[20] M. J. Zaki and W. M. Jr. 2014. *Data Mining and Analysis: Fundamental Concepts and Algorithms.* Cambridge University Press, New York, NY, USA.